Localizing Objects with Self-Supervised Transformers and no Labels

Oriane Siméoni\textsuperscript{1}, Gilles Puy\textsuperscript{1}, {oriane.simeoni,gilles.puy}@valeo.com
Huy V. Vo\textsuperscript{1,2}, Simon Roburin\textsuperscript{1,3} van-huy.vo@inria.fr, simon.roburin@gmail.com
Spyros Gidaris\textsuperscript{1}, Andrei Bursuc\textsuperscript{1}, {spyros.gidaris,andrei.bursuc}@valeo.com
Patrick Pérez\textsuperscript{1}, Renaud Marlet\textsuperscript{1,3}, {patrick.perez,renaud.marlet}@valeo.com
Jean Pérez\textsuperscript{1}, Renaud Marlet\textsuperscript{1,3}, {patrick.perez,renaud.marlet}@valeo.com
1 Valeo.ai, Paris, France
2 Inria and DIENS (ENS-PSL, CNRS, Inria), Paris, France
3 LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France
4 Center for Data Science, New York University, New York, USA

Abstract

Localizing objects in image collections without supervision can help to avoid expensive annotation campaigns. We propose a simple approach to this problem, that leverages the activation features of a vision transformer pre-trained in a self-supervised manner. Our method, LOST, does not require any external object proposal nor any exploration of the image collection; it operates on a single image. Yet, we outperform state-of-the-art object discovery methods by up to 8 CorLoc points on PASCAL VOC 2012. We also show that training a class-agnostic detector on the discovered objects boosts results by another 7 points. Moreover, we show promising results on the unsupervised object discovery task. The code can be found at https://github.com/valeoai/LOST.

Figure 1: Three applications of LOST to unsupervised single-object discovery (left), multi-object discovery (middle) and object detection (right). In the latter case, objects discovered by LOST are clustered into categories, and cluster labels are used to train a classical object detector. Although large image collections are used to train the underlying image representation [13] and the detector [51], no annotation is ever used in the pipeline. See Figure 3 and Tables 1, 3 for more experiments.

1 Introduction

Object detectors are now part of critical systems, such as autonomous vehicles. However, to reach a high level of performance, they are trained on a vast amount of costly annotated data. Various approaches have been proposed to reduce these costs, such as semi-supervision [41], weak supervision [52], active-learning [3] and self-supervision [26] with task fine-tuning.

© 2021. The copyright of this document resides with its authors. It may be distributed unchanged freely in print or electronic forms.
We consider here the extreme case of localizing objects in images without any annotation. Early works investigate regions proposals based on saliency [84] or intra-image similarity [65], i.e., only between patches within the considered image (not across the image collection). However, these proposals have low precision and are produced in large quantities only to reduce the search space in other tasks, such as supervised [27, 28] or weakly-supervised [9, 59] object detection. Often using region proposals as input, unsupervised object discovery leverages information from the entire image collection and explores inter-image similarities to localize objects in an unsupervised fashion, e.g., with probabilistic matching [15], principal component analysis [72], optimization [67, 68] and ranking [69]. However, because of the quadratic complexity of region comparison among images, together with the high number of region proposals for a single image, these methods hardly scale to large datasets. Other approaches do not require annotations but exploit extra modalities, e.g., audio [2] or LiDAR [61].

We propose here a simple approach to localize objects in an image, that we then apply to unsupervised object discovery. Our localization method stays at the level of a single image, rather than exploring inter-image similarity, which makes it linear w.r.t. the number of images and thus highly scalable. For this, we leverage high-quality features obtained from a visual transformer pre-trained with DINO self-supervision [13]. Concretely, we divide the image of interest into equal-sized patches and feed it to the DINO model. Instead of focusing on the CLS token, we propose to use the key component of the last attention layer for computing the similarities between the different patches. In doing so, we are able to localize a part of an object by selecting the patch with the least number of similar patches, here called the seed. The justification for this seed selection criterion is based on the empirical observation that patches of foreground objects are less correlated than patches corresponding to background. We add to this initial seed other patches that are highly correlated to it and thus likely to be part of the same object, a process which we call seed expansion. Finally, we construct a binary object segmentation mask by computing the similarities of each image patch to the selected seed patches and infer the bounding box of an object as the box that tightly encloses the largest connected component in this mask that contains the initial seed. In following this simple method, we not only outperform methods for region proposals but also those for single-object discovery. Even more, by training an off-the-self class-agnostic object detector using our localized boxes as ground-truth boxes, we are able to derive a much more accurate object localization model that is actually able to detect multiple objects in an image. We call this task unsupervised class-agnostic object detection (which may resort to self-supervision despite being called unsupervised). Finally, by using clustering techniques to group the localized objects into visual consistent classes, we are able to train class-aware object detectors without any human supervision, but using instead the predicted object locations and their cluster ids as ground-truth annotations. We call this task unsupervised (class-aware) object detection. We show that the predictions of our unsupervised detection model for certain clusters correlate very well with labelled semantic classes in the dataset and reach for them detection results competitive to object detectors trained with weak supervision [9, 59].

Our main contributions are as follows: (1) we show how to extract relevant features from a self-supervised pre-trained vision transformer and use the patch correlations within an image to propose a simple single-object localization method with linear complexity w.r.t. to dataset size; (2) we leverage it to train both class-agnostic and class-aware unsupervised object detectors able to accurately localize multiple object per image and, in the class-aware case, group them to semantically-coherent classes; (3) we outperform the state of the art in unsupervised object discovery with a significant margin.
2 Related work

Object detection with limited supervision. Region proposal methods [4, 65, 84] generate in an unsupervised way numerous class-agnostic bounding boxes with high recall but low precision, to speed-up sliding window search. From supervised pre-trained networks, objects can emerge by masking the input [7], interpreting neurons [81] or from saliency maps [54]. Weakly-supervised object detection (WSOD) uses image-level labels without bounding boxes [9, 59] to learn to detect objects. The different instances of WSOD (each with specific assumptions on the availability and amount of image-level and box-level annotations) are often addressed as semi-supervised learning [23, 60] and leverage self-training [37, 49]. Recent work replaces manual annotations with automatic supervision from a different modality, e.g., LiDAR [61] or audio [2]. In contrast, we do not use any annotations or other modalities at any stage: we extract object candidates from the activations of a self-supervised pre-trained network, compute pseudo-labels and then train an object detector.

Object discovery. Given a collection of images, object discovery groups images depicting similar objects, and then localizes objects within these images. Early works [29, 53, 56, 58, 71] focus mostly on the first task and to, a lesser extent, on localization [53, 56, 78, 82]. On the contrary, [15, 38, 67, 68, 69] shift focus on the second task and achieve good object localization on image collections in the wild. However, casting object discovery as the selection of recurring visual patterns across an image collection involves expensive computation and only [69] is able to scale to large datasets. Our work also discovers object locations but does not consider inter-image similarity. Instead, we rely on the power of self-supervised transformer features [13] and only consider intra-image similarity. Consequently, our method can localize objects in a single image with little computation. Close to ours, [80] is also able to localize objects from a single image by exploiting scale-invariant features. Finally, some works [10, 20, 30, 43, 46] on object discovery attempt to simultaneously learn an image representation and to decompose images into object masks. These works, however, are only evaluated on image collections of very simple geometric objects.

Transformers. In this work, we leverage transformer representations to address object discovery. Self-attention layers have been previously integrated into CNNs [11, 35, 70], yet transformers for vision are very recent [14, 16, 18, 50] and still in an incipient stage. Findings on training heuristics [62, 77] and architecture design [42, 63, 76] are released at high pace. Early adaptations of transformers to different tasks (e.g., image classification [18], retrieval [19], object detection [11, 42, 83] and semantic segmentation [42, 57, 74]) have demonstrated their utility and potential for vision. Meanwhile, several works attempt to better understand this new family of models from various perspectives [8, 13, 45, 47, 64]. Interestingly, transformers have been shown to be less biased towards textures than CNNs [47, 64], hinting that their features encapsulate more object-aware representations. These findings motivate us to study manners of localizing objects from transformer features.

Self-supervised learning (SSL) is a powerful training scheme to learn useful representations without human annotations. It does so via a pretext learning task for which the supervision signal comes from the data itself [24, 48, 79]. SSL pre-trained networks have been shown to outperform ImageNet pre-trained networks on several computer vision tasks, in particular object detection [12, 25, 26, 31, 34]. For transformers, SSL methods also work well [13, 75], bringing a few interesting side-effects. In particular, DINO [13] feature activations appear to contain explicit information about the semantic segmentation of objects in an image. In the same spirit, we extract another kind of transformer features to build our object localization.
3 Proposed approach

Our method exploits image representations extracted by a vision transformer. In this section, we first recall how such representations are obtained, then present our method.

3.1 Transformers for Vision

Input. Vision transformers operate on a sequence of patches of fixed size $P \times P$. For a color image $I$ of spatial size $H \times W$, we have $N = HW/P^2$ patches of size $3P^2$ (we assume for simplicity that $H$ and $W$ are multiples of $P$). Each patch is first embedded in a $d$-dimensional latent space via a trained linear projection layer. An additional, learned vector called the “class token”, $\text{CLS}$, is adjoined to the patch embeddings, yielding a transformer input in $\mathbb{R}^{(N+1) \times d}$.

Self-attention. Transformers consist of a sequence of multi-head self-attention layers and multi-layer perceptrons (MLPs) \cite{18, 66}. Three different learned linear transformations are applied to an input $X \in \mathbb{R}^{(N+1) \times d}$ of a self-attention layer to produce a query $Q$, a key $K$ and a value $V$, all in $\mathbb{R}^{(N+1) \times d}$. The output of the self-attention layer is $Y = \text{softmax}(d^{-1/2} QK^\top) V \in \mathbb{R}^{(N+1) \times d}$, where softmax is applied row-wise. For simplicity, we describe here the case of a single-head attention layer, but attention layers usually contain multiple heads. In this work, we concatenate the keys (or queries, or values) from all heads in the last self-attention layer to obtain our feature representations.

Features for object localization. We use transformers trained in a self-supervised manner using DINO \cite{13}. Caron et al. \cite{13} show that sensible object segmentations can be obtained from the self-attention of the $\text{CLS}$ query produced by the last attention layer. We adapt this strategy in section 4 to perform object localization, providing a baseline (‘DINO-seg’) that produces fair results. However, we found that its does not fully exploit the potential of the self-supervised transformer features. We propose a novel and effective strategy for localizing objects using another way to extract and use features. Our method, called LOST, is constructed by computing similarities between patches of a single image, using this time patch keys $k_p \in \mathbb{R}^d$, $p = 1, \ldots, N$, extracted at the last layer of a transformer.

3.2 Finding objects with LOST

Our method takes as input $d$-dimensional image features $F \in \mathbb{R}^{N \times d}$ extracted from a single image via a neural network; $N$ denotes the spatial dimension (number of patches) of the image features $F$, while $f_p \in \mathbb{R}^d$ is the feature vector of the patch at spatial position $p \in \{1, \ldots, N\}$. We assume that there is at least one object in the image and LOST tries to localize one of them given the input features. To that end, it relies on a selection of patches that are likely to belong to an object. We call these patches “seeds”.

Initial seed selection. Our seed selection strategy is based on the assumptions that (a) regions/patches within objects correlate more with each other than with background patches and vice versa, and (b) an individual object covers less area than the background. Consequently, a patch with little correlation in the image has higher chances to belong to an object. To compute the patch correlations, we rely on the distinctiveness of self-supervised transformer features, which is particularly noticeable when using transformer’s keys. We empirically observe that using these transformer features as patch representation meets assumption (a) in practice: patches in an object correlate positively with each other but negatively
Figure 2: **Initial seed, patch similarities and patch degrees.** Top: images from Pascal VOC2007. Middle: initial seed $p^*$ (in red) and patches similar to $p^*$ (in grey), i.e., such that $f_p^\top f_q \geq 0$ hence $a_{p^*q} = 1$. Bottom: map of inverse degrees $1/d_p$ of all patches $p$ (yellow to blue, for low to high degrees). The initial seed $p^*$ is the patch with the lowest degree. Figure is best viewed in color.

with patches in the background. Therefore, based on assumption (b), we select the first seed $p^*$ by picking the patch with the smallest number of positive correlations with other patches.

Concretely, we build a patch similarity graph $G$ per image, represented by the binary symmetric adjacency matrix $A = (a_{pq})_{1 \leq p, q \leq N} \in \{0, 1\}^{N \times N}$ such that

$$a_{pq} = \begin{cases} 1 & \text{if } f_p^\top f_q \geq 0, \\ 0 & \text{otherwise}. \end{cases}$$

(1)

In other words, two nodes $p, q$ are connected by an undirected edge if their features $f_p, f_q$ are positively correlated. Then, we select the initial seed $p^*$ as a patch with the lowest degree $d_p$:

$$p^* = \arg\min_{p \in \{1, \ldots, N\}} d_p \quad \text{where} \quad d_p = \sum_{q=1}^{N} a_{pq}. \quad (2)$$

We show in Figure 2 examples of seeds $p^*$ selected in four different images. A representation of the degree map for each of these images is also presented. We remark that the patches with lowest degrees are the most likely to fall in an object. Finally, we also observe in this figure that the few patches that correlate positively with $p^*$ are also likely to belong to an object.

**Seed expansion.** Once the initial seed is selected, the second step consists in selecting patches correlated with the seed that are also likely to fall in the object. Again, we achieve this step relying on the empirical observations that pixels within an object tend to be positively correlated and to have a small degree in $G$. We select the next best seeds after $p^*$ as the pixels that are positively correlated with $f_{p^*}$: $S = \{q \mid q \in D_k \text{ and } f_q^\top f_{p^*} \geq 0\}$ within $D_k$, the $k$ patches with the lowest degree. (In case of patches with equal degrees, we break ties arbitrarily to ensure that $|D_k| = k$.) Note that $p^* \in D_k$ and a typical value for $k$ is 100.
Box extraction. The last step consists in computing a mask $m \in \{0, 1\}^N$ by comparing the seed features in $S$ with all the image features. The $q^{th}$ entry of the mask $m$ satisfies

$$m_q = \begin{cases} 
1 & \text{if } \sum_{s \in S} f_q^T f_s \geq 0, \\
0 & \text{otherwise.}
\end{cases}$$ (3)

In other words, a patch $q$ is considered as part of an object if, on average, its feature $f_q$ positively correlates with the features of the patches in $S$. To remove the last spurious correlated patches, we finally select the connected component in $m$ that contains the initial seed and use the bounding box of this component as the detected object. An illustration of the detected boxes before and after seed expansion is provided in Figure 3.

3.3 Towards unsupervised object detection

We exploit the accurate single-object localization of LOST for training object detection models without any human supervision. Starting from a set of unlabeled images, each one assumed to contain at least one prominent object, we extract one bounding box per image using LOST. Then, we train off-the-shelf object detectors using these pseudo-annotated boxes. We explore two scenarios: class-agnostic and (pseudo) class-aware training of object detectors.

Class-agnostic detection (CAD). A class-agnostic detection model localizes salient objects in an image without predicting nor caring about their semantic category. We train such a detector by assigning the same “foreground” category to all the boxes produced by LOST, which we call “pseudo-boxes” afterwards, as they are obtained with no supervision. Unlike LOST, the trained detector can localize multiple objects per image, even if it was trained on a dataset containing only one pseudo-box annotation per image. The experiments confirm that the trained detector can output multiple detections and the quantitative results (Table 1) show that this trained detector is in fact even better than LOST in terms of localization accuracy.

Class-aware detection (OD). We now consider a typical detector that both localizes objects and recognizes their semantic category. To train such a detector, apart from LOST’s pseudo-boxes, we also need a class label for each of these boxes. In order to remain fully-unsupervised, we discover visually-consistent object categories using K-means clustering. For each image, we crop the object detected by LOST, resize the cropped image to 224 $\times$ 224, feed this image in the DINO pre-trained transformer, and extract the CLS token at the last layer. The set of CLS tokens are clustered using K-means and the cluster index is used as a pseudo-label for training the detector. At evaluation time, we match these pseudo-labels to the ground-truth class labels using the Hungarian algorithm [39], which give names to pseudo-labels.
4 Experiments

We explore in this section three variants of the object localization problem, in order of increasing complexity: (1) localizing one salient object in each image (single-object discovery) in §4.2, (2) using the corresponding bounding boxes as ground-truth to train a binary classifier for foreground object detection (unsupervised class-agnostic object detection), and (3) using clustering to capture an unsupervised notion of object categories, and detect the corresponding instances (unsupervised object detection). Both are discussed in §4.3. None of the building blocks of this pipeline uses any annotation, just a large number of unlabelled images to sequentially train, in a self-supervised way, the DINO transformer, the class agnostic foreground/background classifier, and finally the classifier using the cluster identifier as labels. Also, we provide more qualitative results in supplementary.

4.1 Experimental setup

**Backbone networks.** Unless otherwise specified, we use the ViT-S model introduced in [13], which follows the architecture of DEiT-S [62]. It is trained using DINO [13], with a patch size of \( P = 16 \) and the keys \( K \) (without the entry corresponding to the \( \text{CLS} \) token) of the last layer as input features \( F \), with which we achieve the best results. Results obtained alternatively with the attention, the queries and values are presented and discussed in the supplementary material. For comparison, we also present results using the base version of ViT (ViT-B), ViT-S with a patch size of \( P = 8 \), as well as with features of the last convolutional layer of a dilated ResNet-50 [32] and of a VGG16 [55] pre-trained either following DINO, or in a supervised fashion on Imagenet [17].

**Datasets.** We evaluate the performance of our approach on the three variants of object localization on VOC07 [21] trainval+test, VOC12 [22] trainval and COCO_20K [40, 68]. VOC07 and VOC12 are commonly used benchmarks for object detection [27, 28]. COCO_20k is a subset of the COCO2014 trainval dataset [40], consisting of 19817 randomly chosen images, used as a benchmark in [68]. When evaluating results on the unsupervised object discovery task, we follow a common practice and evaluate scores on the trainval set of the different datasets. Such an evaluation is possible as the task is fully unsupervised. We follow the same principle for the unsupervised class-agnostic task: we generate boxes on VOC07 trainval, VOC12 trainval and COCO_20k, use them to train a class-agnostic detector, and then evaluate again on these datasets (against ground-truth boxes this time). For unsupervised class-aware object detection, we generate boxes and train the detector on VOC07 trainval and/or VOC12 trainval, but evaluate the detector on the VOC07 test set to facilitate comparisons to weakly-supervised object detection methods. Note that for unsupervised object discovery, some previous works [67, 68, 69, 72] evaluate on subsets of VOC07 trainval and VOC12 trainval. For completeness, we present the object discovery performance of our method on these reduced datasets in the supplemental material.

4.2 Application to unsupervised object discovery

Similar to methods for unsupervised single-object discovery, LOST produces one box for each image. It therefore can be directly evaluated for this task. Following [15, 67, 68, 69, 80], we use the Correct Localization (CorLoc) metric, i.e., the percentage of correct boxes, where a predicted box is considered correct if it has an intersection over union (IoU) score superior to 0.5 with one of the labeled object bounding boxes.
Table 1: Single-object discovery. CorLoc performance on VOC07 trainval, VOC12 trainval and COCO_20k. We compare LOST to state-of-the-art object discovery methods [38, 68, 69, 72, 80], as well as to two object proposal methods [65, 84]. We also compare to the segmentation method proposed in DINO [13], denoted by DINO-seg. Additionally, we train a class-agnostic detector (+ CAD) using as ground-truth either our pseudo-boxes or the boxes of rOSD [68] or LOD [69].

Comparison to prior work. In Table 1, we present the CorLoc of our method, in comparison to state-of-the-art object discovery methods [38, 68, 69, 72, 80] and region proposals [65, 84]. Despite its simplicity, we see that LOST outperforms the other methods by large margins. We also compare against an adapted version of the segmentation method proposed in [13]. Concretely, we extract the self-attention of the CLS query at the last layer of the transformer, create a binary mask where the 0.6N largest entries of this self-attention are set to 1, retrieve the largest spatially-connected component from this binary mask, and use the bounding box of this component as the detected object. This method returns one box per self-attention head and we report results obtained with the best performing head over the entire dataset, noted as DINO-seg. LOST improves over DINO-seg by 8 to 17 of CorLoc points, demonstrating the efficacy of our approach for object localization based on self-supervised pre-trained transformer features.

Finally, we also evaluate our unsupervised class-agnostic detector (denoted by ‘+ CAD’) for single-object discovery. To this end, we return for each image the box that the detector assigns the highest score. It can be seen that training a class-agnostic detector on LOST’s outputs further improves the performance by 4 to 7 CorLoc points. In total, our method surpasses the prior state of the art by at least 10 CorLoc points on each evaluated dataset.

Impact of the backbone architecture. Table 2 studies the effect of the backbone on LOST. We see that transformer representations are better suited for our method (best results with ViT-S/16). In contrast, our performance using the DINO-pre-trained ResNet-50 is significantly lower. It indicates that the performance of our method is not only due to the contributions of self-supervision but also to the property and quality of the specific features we extract.

4.3 Unsupervised object detection

Here we explore the application of LOST in unsupervised object detection. To that end, we use LOST’s pseudo-boxes to train a Faster R-CNN model [51] on the datasets. We measure detection performance using the Average Precision at IoU 0.5 metric (AP@0.5), which is commonly used in the PASCAL detection benchmark. As Faster R-CNN backbone, we use a
Table 2: **Impact of the backbone.** We evaluate LOST on features originating from different backbones: ViT [18] small (ViT-S) and base (ViT-B) with patch size $P = 8$ or 16, ResNet50 [32] pre-trained following DINO [13], and VGG16 [55] and ResNet50 trained in a fully-supervised fashion on Imagenet [17].

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Pre-training</th>
<th>VOC07_trainval</th>
<th>VOC12_trainval</th>
<th>COCO_20k</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td>supervised</td>
<td>42.0</td>
<td>47.2</td>
<td>30.2</td>
</tr>
<tr>
<td>ResNet50</td>
<td>supervised</td>
<td>33.5</td>
<td>39.1</td>
<td>25.5</td>
</tr>
<tr>
<td>ResNet50</td>
<td>DINO</td>
<td>36.8</td>
<td>42.7</td>
<td>26.5</td>
</tr>
<tr>
<td>ViT-S/8</td>
<td>DINO</td>
<td>55.5</td>
<td>57.0</td>
<td>49.5</td>
</tr>
<tr>
<td>ViT-S/16</td>
<td>DINO</td>
<td><strong>61.9</strong></td>
<td><strong>64.0</strong></td>
<td><strong>50.7</strong></td>
</tr>
<tr>
<td>ViT-B/16</td>
<td>DINO</td>
<td>60.1</td>
<td>63.3</td>
<td>50.0</td>
</tr>
</tbody>
</table>

Table 3: **Object detection.** Results (AP@0.5 %) on VOC07 test. LOST + OD and rOSD [68] + OD are trained on VOC07 trainval. LOST + OD† is trained on the union of VOC07 and VOC12 trainval sets.

ResNet50 pre-trained with DINO self-supervision, thus making our training pipeline fully-unsupervised. We trained the Faster R-CNN models using the detectron2 [73] implementation (more details in the supplementary material).

**Pseudo-labels.** To generate pseudo-labels for the class-aware detectors, we apply K-means clustering on DINO-ViT-S tokens using as many clusters as the number of different classes in the dataset. Since the cluster-based pseudo-labels are “anonymous”, to evaluate the detection results we must map the clusters to the ground-truth classes. Following prior work in image clustering [5, 6, 36], we use Hungarian matching [39] for that. We stress that this matching is only for reporting evaluation results; we do not use any human labels during training.

**Unsupervised class-aware detection.** Table 3 provides results of unsupervised class-aware object detectors trained with LOST (entry ‘LOST + OD’). We are not aware of any prior work that addresses unsupervised object detection on real-world images of complex scenes, as those in PASCAL, that does not use extra modalities. We could not compare to [1, 61] as we focus on image-only benchmarks.

We see that, although fully-unsupervised, our method learns to accurately detect several object classes. For example, detection performance for classes “aeroplane”, “bus”, “dog”, “horse” and “train” is more than 50.0%, and for “cat” it reaches 72.2%. Even more so, for some classes our method achieves better AP than the weakly-supervised methods WSDDN [9] and PCL [59], which require image-wise human labels. Although the results are not entirely comparable due to backbone differences between our method and the weakly-supervised ones (self-supervised ResNet50 vs. supervised VGG16), they still demonstrate the efficacy of our method in unsupervised object detection, which is an extremely hard and ill-posed task.

We also evaluate the AP of our pseudo-boxes (with their assigned cluster id as pseudo-labels) when generated for VOC07 test (entry ‘LOST pseudo-boxes’). Evidently, training the detector on pseudo-boxes leads to a significantly higher AP than the initial pseudo-boxes.

Finally, switching our pseudo-boxes with those of rOSD [68] for the detector training (adding pseudo-labels to rOSD pseudo-boxes by clustering DINO features in exactly the same way as in our method) leads to performance degradation (entry ‘rOSD + OD†’).

**Unsupervised class-agnostic detection.** In Table 4, we report class-agnostic detection results obtained using pseudo-boxes from our method (‘LOST + CAD’) as well as from
<table>
<thead>
<tr>
<th>Training set (when applicable)</th>
<th>Evaluation set</th>
<th>VOC07</th>
<th>VOC12</th>
<th>COCO20k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>trainval</td>
<td>test</td>
<td>trainval</td>
<td>trainval</td>
</tr>
<tr>
<td>EdgeBoxes [84]</td>
<td>3.6</td>
<td>4.4</td>
<td>4.8</td>
<td>1.8</td>
</tr>
<tr>
<td>Selective Search [65]</td>
<td>2.9</td>
<td>3.6</td>
<td>4.2</td>
<td>1.6</td>
</tr>
<tr>
<td>rOSD [68] + CAD</td>
<td>24.2</td>
<td>25.2</td>
<td>29.0</td>
<td>8.4</td>
</tr>
<tr>
<td>LOD [69] + CAD</td>
<td>22.7</td>
<td>23.7</td>
<td>28.4</td>
<td>8.8</td>
</tr>
<tr>
<td>LOST + CAD</td>
<td>29.0</td>
<td>29.0</td>
<td>33.5</td>
<td>9.9</td>
</tr>
</tbody>
</table>

### Table 4: Class-agnostic unsupervised object detection results (in AP@0.5 %). Trainings, corresponding to ‘method + CAD’, are performed on the bare images and rely only on the fully-unsupervised methods rOSD [68], LOD [69] and LOST (ours). Evaluation of unsupervised object detection may thus be performed on the same images as those used for unsupervised training (without manual annotations). The classic methods EdgeBoxes [84] and Selective Search [65] do not involve any training.

rOSD [68] (‘rOSD + CAD’) and LOD [69] (‘LOD + CAD’). As we see, our method leads to a significantly better detection performance. We also report detection results using the Selective Search [65] and EdgeBox[84] proposal algorithms, which perform worse than our method.

### 4.4 Limitations and future work

Despite the good performance of LOST, it exhibits some limitations.

LOST, as it stands, can separate same-class instances that do not overlap (as it only keeps the connected component of the initial seed to create a box), but it is not designed to separate instances when overlapping. This is actually a challenging problem, related to the difference between supervised semantic [44] and instance [33] segmentation methods, which, as far as we know, is an open problem in the absence of any supervision. A potential lead could be to use a matching algorithm such as Probabilistic Hough Matching to separate instances within image regions found in multiple images.

Another issue is when an object covers most of the image. It violates our second assumption for the initial seed selection (expressed in subsection 3.2) that an individual object covers less area than the background, thus possibly causing the seed to fall in the background instead of a foreground object. Ideally, we would like to filter out such failure cases, e.g., by using the attention maps of the CLS token. We leave this as future work.

### 5 Conclusion

We have presented LOST, a simple, yet effective method for localizing objects in images without any labels, by leveraging self-supervised pre-trained transformer features [13]. Despite its simplicity, LOST outperforms state-of-the-art methods in object discovery by large margins. Having high precision, the boxes found by LOST can be used as pseudo ground truth for training a class-agnostic detector which further improves the object discovery performance. LOST boxes can also be used to train an unsupervised object detector that yields competitive results compared to weakly-supervised counterparts for several classes.

Future work will be dedicated to investigate other applications of LOST boxes, e.g., high-quality region proposals for object detection tasks, and the power of self-supervised transformer features for unsupervised object segmentation.
6 Acknowledgments and Disclosure of Funding

This work was supported in part by the Inria/NYU collaboration, the Louis Vuitton/ENS chair on artificial intelligence and the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute). Huy V. Vo was supported in part by a Valeo/Prairie CIFRE PhD Fellowship.

References


Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR, 2009.


